Lab Task: React Native FlatList

Objective:

Students will learn to implement and customize the FlatList component in React Native. They will display lists using both simple arrays and arrays of objects, and explore the use of key props such as renderItem, keyExtractor, horizontal layout, and multiple columns.

Activity Outcomes:

After completing this lab, students will be able to:

- Understand the purpose and efficiency of FlatList in React Native.
- Display data using FlatList with simple arrays and arrays of objects.
- Customize FlatList with keyExtractor, renderItem, and styles.
- Implement FlatList with headers, footers, and separators.

Instructor Note:

As a pre-lab activity, students should review the React Native documentation on FlatList and understand the difference between ScrollView and FlatList.

1) Useful Concepts

FlatList is a performant interface for rendering large lists in React Native. It renders only the items that are currently visible on the screen, improving performance. FlatList accepts a data array and a renderItem function to define how each element should be displayed.

2) Solved Lab Activities

Activity 1: Displaying a Simple Array

Write a React Native app using FlatList to display a list of fruits stored in a simple array.

Solution:

- ```
  import React from "react";
  import { FlatList, Text, View, StyleSheet } from "react-native";

  export default function App() {
   const fruits = ["Apple", "Banana", "Cherry", "Mango", "Orange"];

   return (
    <View style={styles.container}>
     <FlatList
      data={fruits}
      renderItem={({ item }) => <Text style={styles.item}>{item}</Text>}
      keyExtractor={(item, index) => index.toString()}
  ```
  ```

```
    />
   </View>
  );
}

const styles = StyleSheet.create({
  container: { flex: 1, padding: 20 },
  item: { fontSize: 18, padding: 10, borderBottomWidth: 1 },
});
```

Activity 2: Displaying an Array of Objects

Use FlatList to display a list of students, showing their names and grades. Each student will be represented as an object.

Solution:

- ```
  import React from "react";
  import { FlatList, Text, View, StyleSheet } from "react-native";

  export default function App() {
    const students = [
      { id: "1", name: "Aisha", grade: "A" },
      { id: "2", name: "Kashif", grade: "B" },
      { id: "3", name: "Sara", grade: "A" },
      { id: "4", name: "Ali", grade: "C" },
    ];

    return (
      <View style={styles.container}>
        <FlatList
          data={students}
          keyExtractor={(item) => item.id}
          renderItem={({ item }) => (
            <View style={styles.itemBox}>
              <Text style={styles.name}>{item.name}</Text>
              <Text>Grade: {item.grade}</Text>
            </View>
          )}
        />
      </View>
    );
  }
  ```
  ```

```
const styles = StyleSheet.create({
  container: { flex: 1, padding: 20 },
  itemBox: {
    padding: 10,
    marginVertical: 5,
    borderWidth: 1,
    borderColor: "#ccc",
    borderRadius: 5,
  },
  name: { fontWeight: "bold" },
});
```

3) Graded Lab Tasks

Complete the following tasks and demonstrate them to your instructor.

Lab Task 1

Use FlatList to display a list of products stored as objects (name and price). Style each product in a card-like view.

Lab Task 2

Implement FlatList with two columns to show a grid layout of colors (use a simple array of color names).

Lab Task 3

Create a simple To-Do List App using FlatList. The app should allow users to add, display, and delete tasks dynamically. Use useState for managing the list and TextInput for entering tasks.

Summary

In this lab, students practiced using FlatList to render both simple arrays and arrays of objects. They learned how to use key props, customize list appearance, and improve performance by using React Native's optimized list rendering. As a graded task, students implemented a basic To-Do List application integrating FlatList with React Native state management.